Changes to PBS Job Requests for V100 GPU Resources

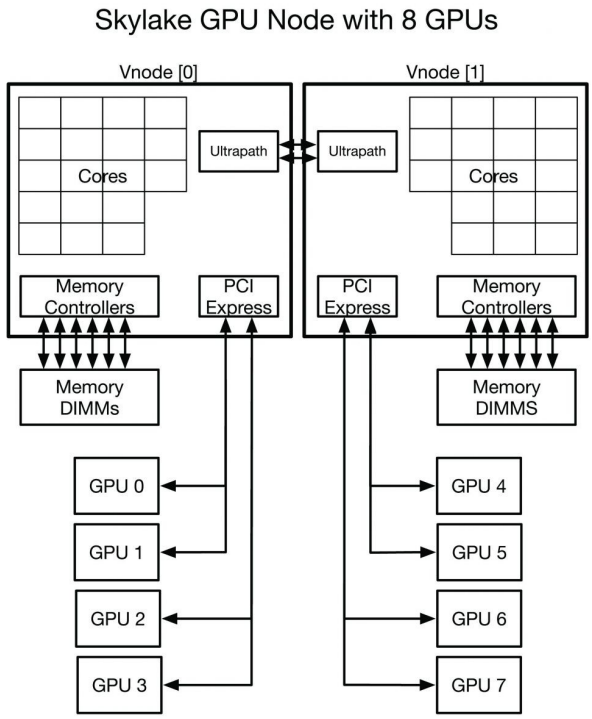
As of July 16, 2020, PBS handles NVIDIA Volta V100 GPU nodes differently than in the past. The new configuration allows multiple PBS jobs to share a single node. This means that you can request and access a subset of GPUs on the node for your job, so each job will utilize only the GPUs it needs, leaving the rest of the GPUs on the node available to other jobs. This enables more efficient use of the V100 nodes.

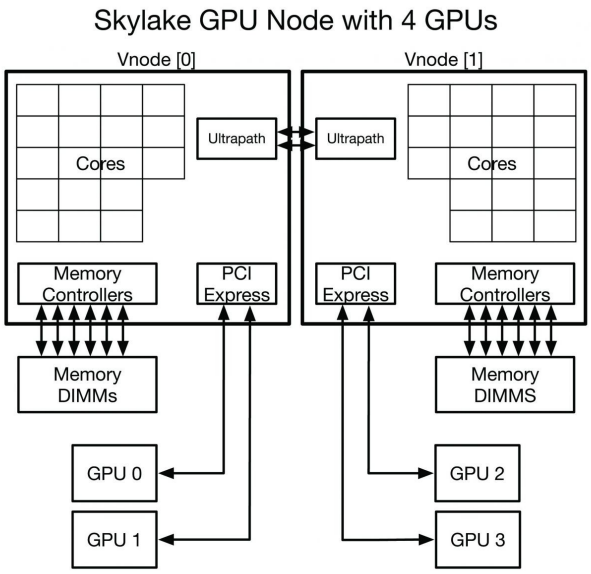
Instead of each node being treated as one unit for exclusive access by a single job, the nodes are logically split into two virtual nodes (*vnodes)*â one for each socket and its associated CPU cores, GPUs, and memory. Unless specified otherwise, a job will have exclusive access only to the resources specified in the job request. Other jobs might run on the same vnodes at the same time but use different CPUs, GPUs and memory.

Note: The **`sky_gpu`** nodes are used below to explain the changes. The explanation also applies to the **`cas_gpu`** nodes except that there are 24 CPU cores per socket in **`cas_gpu`**.

Configuration of Skylake GPU Vnodes

The following diagrams show the logical splitting of four-GPU and eight-GPU Skylake nodes into vnodes. This configuration is similar to the standard Electra Skylake node, except that the GPU version has 18 cores and 192 gigabytes (GB) of memory per socket.





Requesting V100 GPU Resources

As a quick review, the following definitions are paraphrased from the Altair PBS Professional User's Guide:

- A host is any computer. Hosts where jobs run are often called nodes.

- A virtual node, or vnode, is an abstract object representing a set of resources that form a usable part of a machine. This could be an entire host, or a nodeboard or a blade. A single host can be made up of multiple vnodes.
- A chunk specifies the value of each resource in a set of resources that are to be allocated as a unit to a job. It is the smallest set of resources to be allocated to a job. All of a chunk is taken from a single host. One chunk may be broken across vnodes, but all participating vnodes must be from the same host.

When you request non-v100 node types using other PBS queues, the job request looks similar to the following:

```
#PBS -l select=100:model=ivy:ncpus=20
```

where **100** indicates how many "chunks" of resources are requested. Each chunk is assigned to a different node, and the job has exclusive access to all the resources on the node. This is not the case with the new **v100** queue.

Specifying Chunk Resources for the v100 Queue

The chunk resources that must be specified for the **v100** queue are:

ncpus
: The number of cores. Both hyperthreads on each core will be included in the chunk.

ngpus
: The number of GPUs. PBS sets the environment variable CUDA_VISIBLE_DEVICES to a list of the appropriate IDs to access the assigned GPUs.

mpiprocs
: The number of MPI ranks.

ompthreads
: PBS sets this into the OMP_NUM_THREADS environment variable.

mem
: The amount of CPU memory. PBS enforces this limit.

Specifying the Place Statement

In addition, you must specify the **place** statement. For our purposes, this statement has the form **place=*arrangement:sharing.***

Arrangement Attribute

Possible values for the ***arrangement*** attribute:

free
: Place job chunks on any vnodes.

pack
: All chunks will be taken from one host.

scatter
: Only one chunk is taken from any host.

vscatter
: Only one chunk is taken from any vnode.

Sharing Attribute

Possible values for the ***sharing*** attribute:

excl
: Only this job uses the vnodes chosen.

exclhost
: The entire host is allocated to the job.

shared
: This job can share the vnodes chosen.

Resource Request Examples

For example, suppose in the past you wanted four v100 nodes all to yourself. You probably specified something like:

```
#PBS -l select=4:model=sky_gpu:naccelerators=4
```

The equivalent new request is:

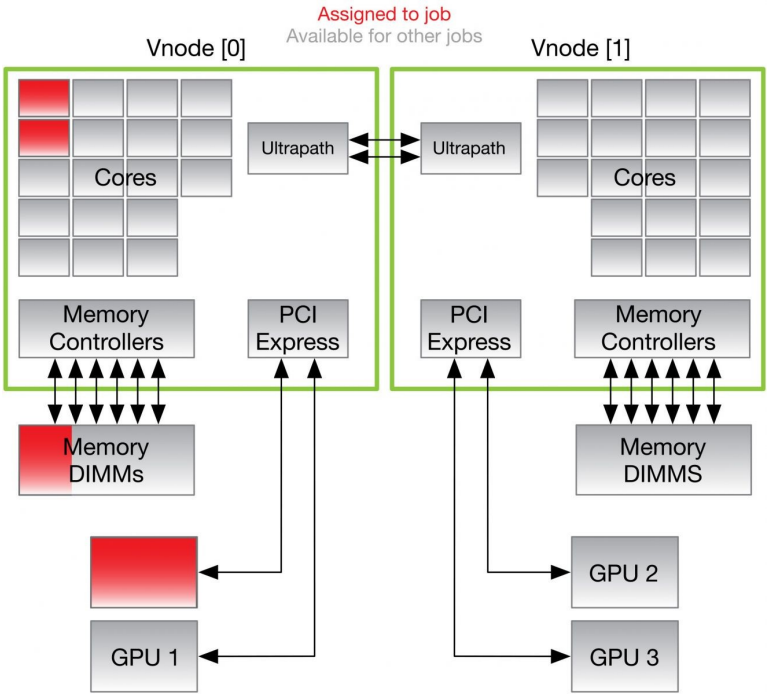
```
#PBS -l select=4:model=sky_gpu:mpiprocs=1:ncpus=36:ngpus=4:mem=300g
#PBS -l place=scatter:excl
```

The number of chunks (4) and the model type (**sky_gpu**) stay the same. However, in order to get access to all the cores, GPUs, and memory on the nodes, you must specify these resources explicitly. Furthermore, on the **place** statement, you must specify exclusive access (**excl**).

Now suppose you want to use just one GPU for a non-MPI program, where the program uses two OpenMP threads on two cores and 40 GB of memory. To make this request:

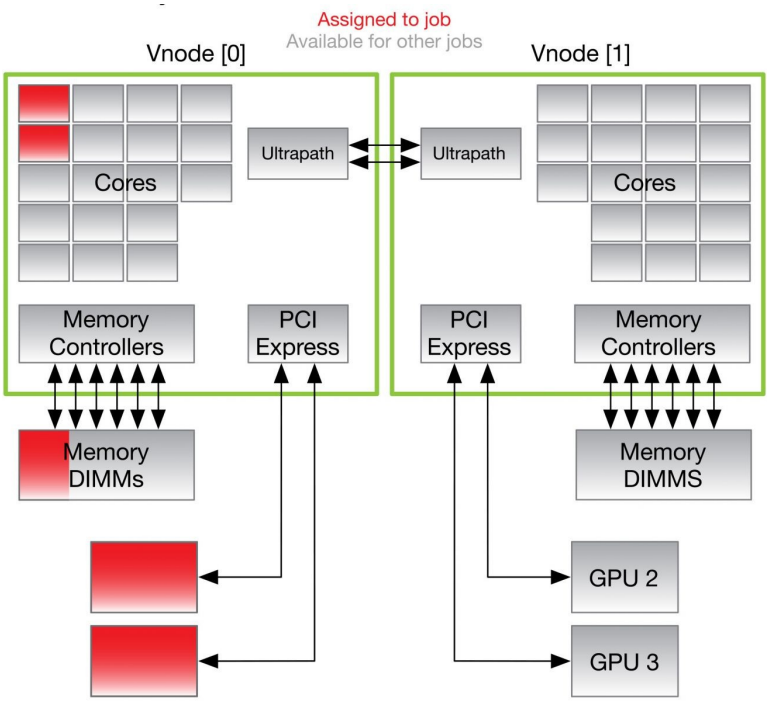
```
#PBS -l select=1:model=sky_gpu:ngpus=1:ncpus=2:ompthreads=2:mem=40g
#PBS -l place=vscatter:shared
```

Because the program is non-MPI, you don't need to specify **mpiprocs**. The other resource values come directly from your requirements. The resulting assignment is shown in the following diagram:



Note: The particular cores, GPUs, and memory might differ, but all resources will come from the same vnode. (Without **vscatter**, the resources might be split across both vnodes in a host.)

To give this program access to two GPUs, just change **ngpus=1** to **ngpus=2**. This results in the following assignment:

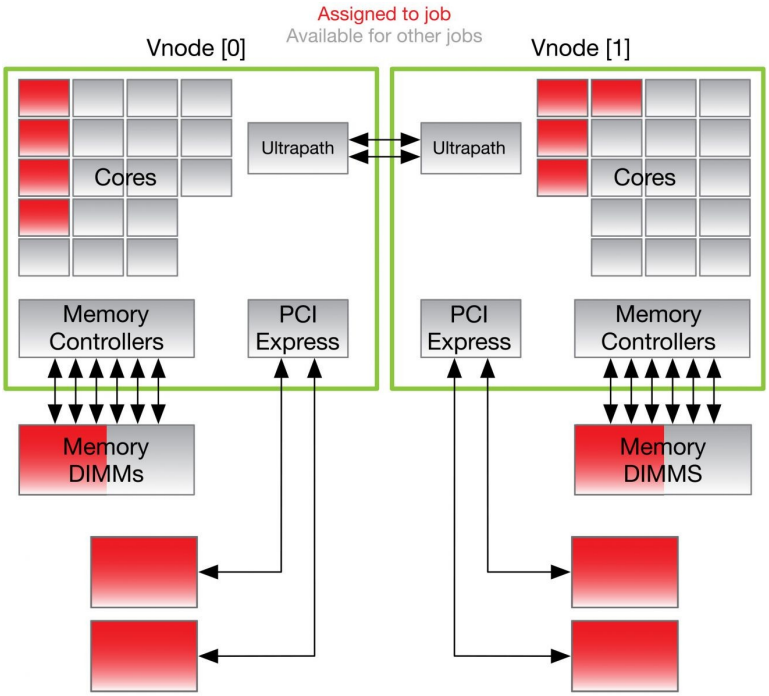


Because these resources comprise significantly less than a whole **sky_gpu** node, the **place** statement indicates that the job can share its node with other jobs.

Suppose, instead, you have an MPI program with four ranks where each rank uses one GPU, two CPU cores, and 50 GB of memory:

```
#PBS -l select=4:model=sky_gpu:ngpus=1:mpiprocs=1:ncpus=2:ompthreads=2:mem=50g
#PBS -l place=pack:shared
```

This results in the following assignment:

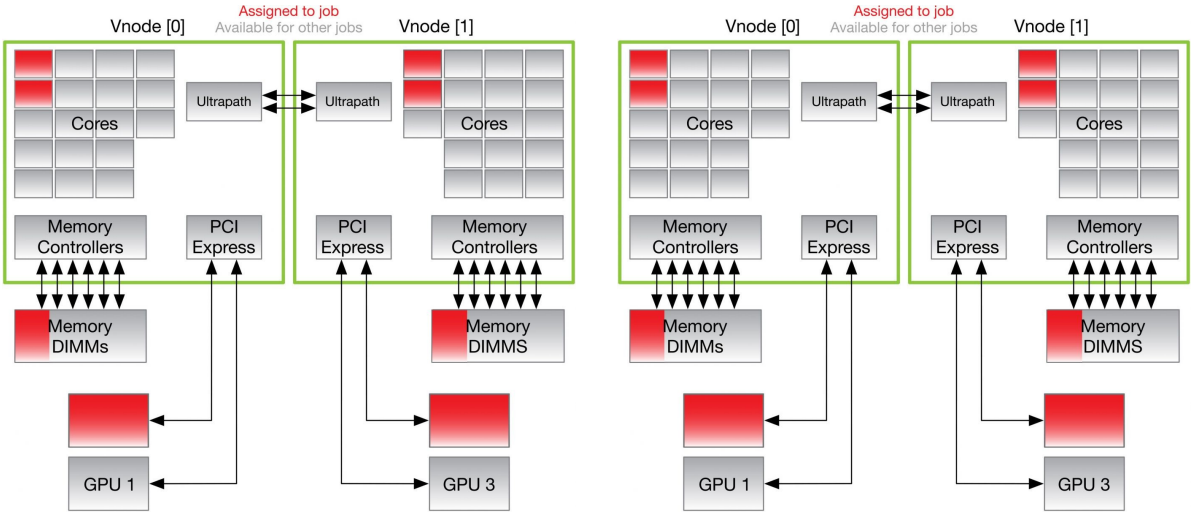


Here, the **`place`** value of **`pack`** means all the chunks will come from the same host, which usually gives the best MPI performance. On the other hand, if MPI is only a small part of the job, you might use **`place=free:shared`** so you don't need to wait for all resources to be available on a single hostâ the resources can come from multiple hosts, wherever they are available.

You might want the GPUs to be on different sockets. In this case, change the **`place`** specification to:

```
#PBS -l place=vscatter:shared
```

Each of the four chunks will be placed on a different vnode, which means the chunks' GPUs will be attached to different sockets. The **`sky_gpu`** nodes have only two sockets, so the job will be spread over multiple hosts:



For more information, see Requesting GPU Resources.

If you have any questions, please contact the NAS Control Room at (800) 331-8737, (650) 604-4444, or support@nas.nasa.gov.

Article ID: 645
Last updated: 18 Dec, 2020
Revision: 53
Systems Reference -> GPU Nodes -> Changes to PBS Job Requests for V100 GPU Resources
https://www.nas.nasa.gov/hecc/support/kb/entry/645/